UNITED STATES PATENT AND TRADEMARK OFFICE

MN

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/672,183	09/25/2003	Eduard K. de Jong	SUN-040024	9848

24209 7590 07/02/2007
GUNNISON MCKAY & HODGSON, LLP
1900 GARDEN ROAD
SUITE 220
MONTEREY, CA 93940

EXAMINER	
CHEN, QING	
ART UNIT	**PAPER NUMBER**
2191	
MAIL DATE	**DELIVERY MODE**
07/02/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/672,183	**Applicant(s)** DE JONG, EDUARD K.	
	Examiner Qing Chen	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 May 2007*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-48* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11 May 2007* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *20070507*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This Office action is in response to the amendment filed on May 11, 2007.

2. **Claims 1-48** are pending.

3. **Claims 1-4, 7-9, 12-15, 18-20, 23-26, 29-31, 34-37, 40-42, 45, and 48** have been amended.

4. The objections to the drawings are withdrawn in view of Applicant's amendments to the drawings and the specification.

5. The objections to the specification are withdrawn in view of Applicant's amendments to the specification.

6. The objections to Claims 3-6, 14-17, 25-28, 36-39, 45, and 48 are withdrawn in view of Applicant's amendments to the claims.

7. The 35 U.S.C. § 112, second paragraph, rejections of Claims 1-48 are withdrawn in view of Applicant's arguments and amendments to the claims.

8. The 35 U.S.C. § 101 rejections of Claims 23-33 are withdrawn in view of Applicant's arguments and amendments to the claims. The 35 U.S.C. § 101 rejections of Claims 41-48 are withdrawn in view of Applicant's amendments to the claims and the Office's current policies regarding non-statutory subject matter.

Response to Amendment

Drawings

9. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description:

- Reference number "3100" in Figure 31.

Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application.

The drawings are objected to because the word "Stream" is misspelled in Figure 32, Element 3250. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application.

Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

10. **Claims 1-7, 12-18, 23-29, 34-40, 46, and 47** are objected to because of the following informalities:

- **Claims 1, 12, 23, and 34** recite the limitation "the opcode value." Applicant is advised to change this limitation to read "the at least one instruction opcode value" for the purpose of providing it with proper explicit antecedent basis.
- **Claims 2-7** depend on Claim 1 and, therefore, suffer the same deficiency as Claim 1.
- **Claims 13-18** depend on Claim 12 and, therefore, suffer the same deficiency as Claim 12.
- **Claims 24-29** depend on Claim 23 and, therefore, suffer the same deficiency as Claim 23.
- **Claims 35-40** depend on Claim 34 and, therefore, suffer the same deficiency as Claim 34.
- **Claims 46 and 47** contain a typographical error: "The memory" should read -- The data processing system --.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

12. **Claims 1, 2, 4-6, 8, 9, 11-13, 15-17, 19, 20, 22-24, 26-28, 30, 31, 33-35, 37-39, 41, 42, and 44-48** are rejected under 35 U.S.C. 102(b) as being anticipated by **Granger et al. (US 6,334,189)**.

As per **Claim 1**, Granger et al. disclose:

- receiving an obfuscated application program, said obfuscated application program comprising at least one instruction opcode value encoded using one of a plurality of instruction set opcode value encoding schemes *(see Column 13: 15-21, "It is therefore preferable to implement each of these components at least in-part either in pseudocode or in obfuscated machine code. In general, only one of these two techniques (pseudocode or obfuscation) will be used to hide the details of a given software function, although both techniques may be (and preferably are) used within the same application." and 28-31, "... the details of the pseudocode instruction set (including the opcodes and instruction formats) are maintained in secrecy by the software developer." and 52-59, "... the pseudocode for performing a given function (or possibly multiple functions) is preferably stored as an encrypted pseudocode ("ECODE") data block 56 within a data table 58 of the executable application file 60. Because the pseudocode is* *stored within a data table, the pseudocode appears to the pirate simply as part of the application's data, and does not impair the operation of the pirate's disassembler or other analysis tool.")*;

- receiving an application program instruction corresponding to a current instruction counter value *(see Column 13: 66-67 through Column 14: 1-7, "... the ECODE data block 56 preferably includes a header 66, any data 68 that is needed by the pseudocode, and the pseudocode instructions, all of which are stored in encrypted, binary form. The header includes information that is used by the interpreter 63 to process the ECODE data block 56, including an initial program counter setting of the emulated CPU, information about the arguments to be passed, and key information for decrypting the instructions 70.")*;

- selecting an instruction dispatch table based at least on said current instruction counter value *(see Column 17: 22-30, "As the sequence of tokens for a given line is read, the token reader matches the opcode to the corresponding instruction in the internal data structure to determine the instruction format and sign information. The token reader then parses the tokens, and maps the tokens (using the mapping macros) into the 32-bit pseudocode instruction. The pseudocode instruction is then written (in unencrypted form) to an instruction list which eventually becomes part of the ECODE data block.")*; and

- executing said application program instruction using said selected instruction dispatch table to obtain a reference to an instruction implementation method corresponding to the at least one instruction opcode value of the said application program instruction *(see Column 18: 52-58, "... the SPEC executes the instruction and updates the PC (block 144). If the instruction involves a retrieval of data from the ECODE data block 56, the data is decrypted using the same* *location-based method used to decrypt instructions. Likewise, if the instruction requires data to be written to the ECODE data block, this data is initially encrypted using the location based-method.").*

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Granger et al. further disclose:

- determining whether there is another application program instruction to be executed *(see Column 18: 63-65, "For all instructions other than branch and jump instructions, the PC is incremented by one to point to the next line of the ECODE data block.")*;

- advancing said current instruction counter if there is another application program instruction to be executed *(see Column 18: 63-65, "For all instructions other than branch and jump instructions, the PC is incremented by one to point to the next line of the ECODE data block.")*; and

- repeating said receiving said application program instruction, said selecting and said executing after said advancing *(see Column 18: 35-39, "Once the header has been decoded, the SPEC loads the registers (block 132) with any arguments and loads the PC with the line number of the first instruction to be fetched and executed. The SPEC then enters into a main fetch/execution loop (blocks 136-144).").*

As per **Claim 4**, the rejection of **Claim 1** is incorporated; and Granger et al. further disclose:

- wherein the number of instruction dispatch tables is based at least on a number of instructions in a method of said obfuscated application program *(see Column 16: 39-48, "... the EASM operates generally by reading one line of the text file (block 100), parsing the line (block 102), processing the parsed line to add data to a set of lists (header, data and code) that eventually become the ECODE data block (blocks 104-112), and then reading the next line of the file (block 100). After all of the lines of the text file have been processed, the EASM merges and encrypts the header, data and code lists (blocks 118 and 120) to generate the ECODE data block 56.").*

As per **Claim 5**, the rejection of **Claim 4** is incorporated; and Granger et al. further disclose:

- wherein said number of instruction dispatch tables is greater than or equal to said number of instructions *(see Column 16: 39-48, "... the EASM operates generally by reading one line of the text file (block 100), parsing the line (block 102), processing the parsed line to add data to a set of lists (header, data and code) that eventually become the ECODE data block (blocks 104-112), and then reading the next line of the file (block 100). After all of the lines of the text file have been processed, the EASM merges and encrypts the header, data and code lists (blocks 118 and 120) to generate the ECODE data block 56.").*

As per **Claim 6**, the rejection of **Claim 5** is incorporated; and Granger et al. further disclose:

- wherein said number of instruction dispatch tables equals said number of instructions *(see Column 16: 39-48, "... the EASM operates generally by reading one line of the text file (block 100), parsing the line (block 102), processing the parsed line to add data to a set of lists (header, data and code) that eventually become the ECODE data block (blocks 104-112), and then reading the next line of the file (block 100). After all of the lines of the text file have been processed, the EASM merges and encrypts the header, data and code lists (blocks 118 and 120) to generate the ECODE data block 56.").*

As per **Claim 8**, Granger et al. disclose:

- reading an application program comprising code *(see Column 16: 5-7, "... the EASM assembler 82 receives as its input a text file 80 which contains assembly code written in the EASM assembly language.")*;

- determining a plurality of dispatch tables associated with said application program *(see Column 17: 22-30, "As the sequence of tokens for a given line is read, the token reader matches the opcode to the corresponding instruction in the internal data structure to determine the instruction format and sign information. The token reader then parses the tokens, and maps the tokens (using the mapping macros) into the 32-bit pseudocode instruction. The pseudocode instruction is then written (in unencrypted form) to an instruction list which eventually becomes part of the ECODE data block.")*;

- transforming said application program into application program code configured to utilize said plurality of dispatch tables during application program execution to determine the location of instruction implementation methods to be executed based at least on using a current instruction counter value to select a dispatch table in said plurality of dispatch tables for use with an application program instruction corresponding to said current instruction counter value *(see Column 16: 58-62, "... if the EASM detects that the line includes an instruction, the EASM's line parser generates a sequence of numeric tokens (3 for most instructions) ... "; Column 17: 22-30, "As the sequence of tokens for a given line is read, the token reader matches the opcode to the corresponding instruction in the internal data structure to determine the instruction format and sign information. The token reader then parses the tokens, and maps the tokens (using the mapping macros) into the 32-bit pseudocode instruction. The pseudocode instruction is then written (in unencrypted form) to an instruction list which eventually becomes part of the ECODE data block. ")*; and

- sending said application program code *(see Column 24: 29-31, "... the workstations 202 may, for example, be in the form of network computers that download copies of the application just prior to execution. ").*

As per **Claim 9**, the rejection of **Claim 8** is incorporated; and Granger et al. further disclose:

- wherein said determining further comprises determining the encoding of said plurality of dispatch tables based at least on a relative frequency of instructions in said application program code *(see Column 17: 8-12, "The internal data structure defines the instruction set of the EASM, and specifies which of the five instruction formats is to be used to encode the assembly language instruction into a 32-bit pseudocode instruction. ").*

As per **Claim 11**, the rejection of **Claim 8** is incorporated; and Granger et al. further disclose:

- after said transforming, applying a cryptographic process to said application program code together with a cryptographic key to create an encrypted obfuscated application program *(see Column 24: 32-35, "The server 200 runs a license management server program 208 ("LM server") that dispatches encrypted authorization certificates to the workstations 202 in response to requests generated by the application 206.")*; and
- said sending comprises sending said encrypted obfuscated application program *(see Column 25: 10-12, "Upon receiving the authorization certificate at the workstation 202, the LM client 210 decrypts and decodes the certificate to ensure that the certificate is valid.").*

Claims 12, 13, and 15-17 are program storage device claims corresponding to the method claims above (Claims 1, 2, and 4-6) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1, 2, and 4-6.

Claims 19, 20, and 22 are program storage device claims corresponding to the method claims above (Claims 8, 9, and 11) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 8, 9, and 11.

Claims 23, 24, and 26-28 are apparatus claims corresponding to the method claims above (Claims 1, 2, and 4-6) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1, 2, and 4-6.

Claims 30, 31, and 33 are apparatus claims corresponding to the method claims above (Claims 8, 9, and 11) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 8, 9, and 11.

Claims 34, 35, and 37-39 are apparatus claims corresponding to the method claims above (Claims 1, 2, and 4-6) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1, 2, and 4-6.

Claims 41, 42, and 44 are apparatus claims corresponding to the method claims above (Claims 8, 9, and 11) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 8, 9, and 11.

As per **Claim 45**, Granger et al. disclose:

- a processor *(see Column 4: 15-17, "An ESD can alternatively be implemented, for example, as a chip on the motherboard of the computer, or as unit of the computer's microprocessor.")*; and
- memory, coupled to said processor, for storing data for access by an application program being executed on said data processing system *(see Column 6: 27-30, "The user data may be encrypted, for example, when the data is written to RAM (random access memory) or mass storage, and may be decrypted when the data is subsequently retrieved or accessed by the application.")*, said memory comprising:
- a data structure stored in said memory, said data structure including information used by said application program to execute an obfuscated application program on said data processing system, said data structure comprising application program code configured to utilize

a plurality of dispatch tables during execution of said obfuscated application program to determine a location of instruction implementation methods to be executed based at least on using a current instruction counter value to select a dispatch table in said plurality of dispatch tables for use with an application program instruction corresponding to said current instruction counter value *(see Column 16: 58-62, "... if the EASM detects that the line includes an instruction, the EASM's line parser generates a sequence of numeric tokens (3 for most instructions) ... "; Column 17: 8-12, "The internal data structure defines the instruction set of the EASM, and specifies which of the five instruction formats is to be used to encode the assembly language instruction into a 32-bit pseudocode instruction." and 20-21, "A similar data structure is used by the SPEC to process the instructions." and 22-30, "As the sequence of tokens for a given line is read, the token reader matches the opcode to the corresponding instruction in the internal data structure to determine the instruction format and sign information. The token reader then parses the tokens, and maps the tokens (using the mapping macros) into the 32-bit pseudocode instruction. The pseudocode instruction is then written (in unencrypted form) to an instruction list which eventually becomes part of the ECODE data block."; Column 18: 22-24, "In operation, the SPEC decrypts and processes an ECODE data block (generated by the EASM) that is stored in the computer's local memory." and 52-58, "... the SPEC executes the instruction and updates the PC (block 144). If the instruction involves a retrieval of data from the ECODE data block 56, the data is decrypted using the same location-based method used to decrypt instructions. Likewise, if the instruction requires data to be written to the ECODE data block, this data is initially encrypted using the location based-method.").*

As per **Claim 46**, the rejection of **Claim 45** is incorporated; and Granger et al. further disclose:

- wherein said data structure further comprises a cryptographic key and protected data, said protected data encrypted using said cryptographic key *(see Column 17: 44-56, "... the SPEC decrypts and executes the instructions line-by-line. It is therefore desirable to use a relatively simple encryption algorithm to encrypt the instructions and data, since the use of a more complex algorithm would reduce instruction throughput. The algorithm used for this purpose is preferably a simple XOR algorithm that uses a key value stored in the header 66 and the position of the data/instruction line in the ECODE list. As indicated above, the key value is generated automatically by the EASM.")*.

As per **Claim 47**, the rejection of **Claim 45** is incorporated; and Granger et al. further disclose:

- wherein said data structure further comprises an obfuscation descriptor that indicates an obfuscation method used to create said obfuscated application program *(see Column 15: 15-19, "... these tools (and the interpreter 62) can be written such that software developers can freely modify the details (opcodes, instruction formats, encryption methods, etc.) of the pseudocode so that the tools can be made publicly available.")*.

As per **Claim 48**, Granger et al. disclose:

- a processor *(see Column 4: 15-17, "An ESD can alternatively be implemented, for example, as a chip on the motherboard of the computer, or as unit of the computer's microprocessor.")*; and

- memory, coupled to said processor, for storing data for access by an application program being executed on said data processing system *(see Column 6: 27-30, "The user data may be encrypted, for example, when the data is written to RAM (random access memory) or mass storage, and may be decrypted when the data is subsequently retrieved or accessed by the application.")*, said memory comprising:

- a data structure stored in said memory, said data structure including information used by said application program to execute an obfuscated application program, said data structure comprising a plurality of dispatch tables used during execution of said obfuscated application program to determine a location of instruction implementation methods to be executed based at least on using a current instruction counter value to select a dispatch table in said plurality of dispatch tables for use with an application program instruction corresponding to said current instruction counter value *(see Column 16: 58-62, "... if the EASM detects that the line includes an instruction, the EASM's line parser generates a sequence of numeric tokens (3 for most instructions) ..."; Column 17: 8-12, "The internal data structure defines the instruction set of the EASM, and specifies which of the five instruction formats is to be used to encode the assembly language instruction into a 32-bit pseudocode instruction." and 20-21, "A similar data structure is used by the SPEC to process the instructions." and 22-30, "As the sequence of tokens for a given line is read, the token reader matches the opcode to the corresponding instruction in the internal data structure to determine the instruction format and sign* *information. The token reader then parses the tokens, and maps the tokens (using the mapping macros) into the 32-bit pseudocode instruction. The pseudocode instruction is then written (in unencrypted form) to an instruction list which eventually becomes part of the ECODE data block."; Column 18: 22-24, "In operation, the SPEC decrypts and processes an ECODE data block (generated by the EASM) that is stored in the computer's local memory." and 52-58, "... the SPEC executes the instruction and updates the PC (block 144). If the instruction involves a retrieval of data from the ECODE data block 56, the data is decrypted using the same location-based method used to decrypt instructions. Likewise, if the instruction requires data to be written to the ECODE data block, this data is initially encrypted using the location based-method.").*

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. **Claims 3, 14, 25, and 36** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Granger et al. (US 6,334,189)** in view of **Folmsbee (US 6,308,256)**.

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and Granger et al. further disclose:

- selecting the instruction dispatch table associated with the result of said modulo-n arithmetic operation *(see Column 17: 33-40, "Once the last line of the text file has been processed, the EASM generates and writes the X-REF and DIS files (block 116), and concatenates the header, data and instruction lists to form a single ECODE list (not shown).")*.

However, Granger et al. do not disclose:

- performing modulo-n arithmetic operation on said current instruction counter value, where n is the number of dispatch tables, each of said dispatch tables associated with a unique number between 0 and n-1.

Folmsbee discloses:

- performing modulo-n arithmetic operation on said current instruction counter value, where n is the number of dispatch tables, each of said dispatch tables associated with a unique number between 0 and n-1 *(see Column 8: 12-13, "... the program counter may increment by amounts from 2 to 18 (modulo 9*4).")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Folmsbee into the teaching of Granger et al. to include performing modulo-n arithmetic operation on said current instruction counter value, where n is the number of dispatch tables, each of said dispatch tables associated with a unique number between 0 and n-1. The modification would be obvious because one of ordinary skill in the art would be motivated to load address locations for instructions into memory, which conform to the program counter incrementation plan *(see Folmsbee – Column 8: 5-9)*.

Claims 14, 25, and 36 are rejected for the same reason set forth in the rejection of Claim 3.

15. **Claims 7, 18, 29, and 40** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Granger et al. (US 6,334,189)**.

As per **Claim 7**, the rejection of **Claim 1** is incorporated; however, Granger et al. do not disclose:

- wherein the number of instruction dispatch tables is based at least on an amount of available memory.

Official Notice is taken that it is old and well known within the computing art to correlate the size of data with the amount of available memory. Memory components are constantly being monitored by a computer system, so that appropriate actions can be taken in the event that portions of the memory become unavailable. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the number of instruction dispatch tables is based at least on an amount of available memory. The modification would be obvious because one of ordinary skill in the art would be motivated to prevent data loss.

Claims 18, 29, and 40 are rejected for the same reason set forth in the rejection of Claim 7.

16. **Claims 10, 21, 32, and 43** rejected under 35 U.S.C. 103(a) as being unpatentable over **Granger et al. (US 6,334,189)** in view of **Chen (US 5,913,064).**

As per **Claim 10**, the rejection of **Claim 8** is incorporated; however, Granger et al. do not disclose:

- wherein said determining further comprises filtering said plurality of dispatch tables to flatten the frequency distribution of instructions over said transformed application program code.

Chen discloses:

- wherein said determining further comprises filtering said plurality of dispatch tables to flatten the frequency distribution of instructions over said transformed application program code *(see Column 5: 61-67, "A function determines whether filtering is performed depending upon the number of operands specified either in the source operand stack or source object fields. The greater the number of source operands, the less likely the instruction will be "filtered". At step 34, if the instruction is "filtered" out in this manner, the instruction is not stored, and a new instruction is generated at step 26.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Chen into the teaching of Granger et al. to include wherein said determining further comprises filtering said plurality of dispatch tables to flatten the frequency distribution of instructions over said transformed application program code. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize generation of instructions, which will most easily pass typechecking, since these instructions will be "over-generated" *(see Chen – Column 5: 55-59)*.

Claims 21, 32, and 43 are rejected for the same reason set forth in the rejection of Claim 10.

Response to Arguments

17. Applicant's arguments filed on May 11, 2007 have been fully considered, but they are not persuasive.

In the remarks, Applicant argues that:

a) The rejection as quoted above failed to cite any teaching of a selection of any table in Granger and instead relied upon mapping macros of Granger. Mapping macros fails to suggest or disclose anything about a table.

Further, there is no teaching that a mapping macro is selected based upon a current value of an instruction counter. Instead, a token in the line of the instruction of Granger is used to select the mapping macro. Accordingly, even if a justification could be made for interpreting mapping macros as a table, the rejection failed to cite any teaching of selecting such a mapping macro based on a current value of an instruction counter and instead relied upon a token selection.

Examiner's response:

a) Examiner disagrees with Applicant's assertion that mapping macros fail to suggest or disclose anything about a table. A mapping macro is interpreted as a table under the broadest reasonable treatment, since both a mapping macro and a table define relationships between various entities.

Examiner also disagrees with Applicant's assertion that there is no teaching that a mapping macro is selected based upon a current value of an instruction counter. Note that the tokens used to select the mapping macro are derived from the instruction *(see Column 16: 58-62, "... if the EASM detects that the line includes an instruction, the EASM's line parser generates a sequence of numeric tokens (3 for most instructions) ... ").*

Note that Applicant did not traverse the Examiner's assertion of Official Notice with regard to Claims 7, 18, 29, and 40. Therefore, the "old and well known within the computing art" statement is taken to be admitted prior art because Applicant has failed to traverse the Examiner's assertion of Official Notice (see MPEP § 2144.03).

Conclusion

18. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / QC
June 23, 2007

WEI ZHEN
SUPERVISORY PATENT EXAMINER